

FACE DETECTION IN CURVELET DOMAIN

MATHEW FRANCIS



DEPARTMENT OF ELECTRICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA
ROURKELA, INDIA

May 2014

FACE DETECTION IN CURVELET DOMAIN

A THESIS SUBMITTED IN PARTIAL FULFILMENT
OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF TECHNOLOGY

IN

ELECTRONIC SYSTEMS AND COMMUNICATION

BY

MATHEW FRANCIS

ROLL NO: 212EE1200

UNDER GUIDANCE OF

DR. (PROF). SUPRATIM GUPTA



DEPARTMENT OF ELECTRICAL ENGINEERING
NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA
ROURKELA, INDIA

May 2014

Declaration of Authorship

I, Mathew Francis, declare that this thesis titled, “ Face Detection in Curvelet Domain” and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a research degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:



NATIONAL INSTITUTE OF TECHNOLOGY ROURKELA

CERTIFICATE

This is to certify that the thesis entitled, “ **Face Detection in Curvelet Domain**” submitted by **Mathew Francis** in partial fulfillment of the requirements for the award of Master of Technology Degree in **Electrical Engineering** with specialization in **Electronic Systems and Communication** during 2013-2014 at the National Institute of Technology, Rourkela (Deemed University) is an authentic work carried out by him under my supervision and guidance.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Date:

Dr.(Prof). Supratim Gupta
Dept. of Electrical Engineering
National Institute of Technology
Rourkela-769008
Odisha, India

Abstract

Face Detection in Curvelet Domain

Face detection and face recognition are two techniques in the field of image processing which have undergone significant research during the past few years. It stems from the availability of powerful algorithms, hardware and the wide range of applications they have got. Though powerful hardware and algorithms are available, today's face detection systems are far from perfect since they work within certain constraints. The performance of a face detection system is affected by the factors such as illumination, pose, occlusion etc. An efficient algorithm is the one which takes into account all the above factors, which would in turn increase the time complexity. Since time complexity acts as the bottleneck, development of an algorithm which detects face in minimum time is the need of the hour. Computations will take less time if a sparse representation can be provided for the image. Curvelet transform is an analysis tool which has the ability to sparsely represent images with curve discontinuities. In this work, curvelet transform is studied and is used to represent face images. Principal component analysis is done on this representation to reduce the dimension of the data. Euclidean distance is the parameter used to classify the face from non-faces. The performance of the system is analyzed using receiver operating characteristics(ROC).

Keywords: Detection, Classifier, Dimension Reduction, Features, Curvelet transform, Principal components

Acknowledgements

Foremost, I would like to express my gratitude to Prof. Supratim Gupta for being an outstanding advisor and excellent professor. His constant encouragement, support, and invaluable suggestions made this work successful.

I thank my fellow lab mates in Embedded System & Real Time Lab: Ram Prabhakar, Sankarasrinivasan, Sajith Kumar, Sreejith Markkassery, Sushant Kumar Panigrahi and Zefree Lazarus, for the intellectually stimulating discussions, for the technical help they have provided, and many funny interactions we had at the Lab.

I am deeply and forever indebted to my parents for their love, support and encouragement throughout my life.

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	iv
Contents	v
List of Figures	vii
List of Tables	viii
Abbreviations	ix
1 Introduction	1
1.1 Motivation & Objective	2
1.2 Factors affecting detection	3
1.3 Applications of Face Detection	3
1.4 Developments in face detection	4
1.5 Organization of the thesis	5
2 Feature Based Face Detection	7
2.1 Literature Survey	7
2.2 Feature based detection	8
2.3 Different methods of detecting a face	8
2.3.1 Overview of face detection system	9
2.3.2 Training Phase	10
2.3.2.1 Obtaining training data	10
2.3.2.2 Define features	11
2.3.2.3 Define Classifier	12
2.3.3 Testing Phase	12
2.4 Possible errors in face detection	12
2.5 Performance comparison	13
2.6 Matlab output for Cascaded Object Detector	14

3	Face Image in Curvelet Domain	15
3.1	Sparse signal	16
3.2	Selection of a transform	16
3.2.1	Wavelet Transform	17
3.2.2	Ridgelet and Curvelet Transform	18
3.3	Fast Discrete Curvelet Transform (FDCT)	19
3.3.1	Curvelet	19
3.3.1.1	Curvelet categories	20
3.3.2	Digital Curvelet Transform	20
3.3.3	Digital Curvelet Transform via wrapping	22
3.3.3.1	Wrapping	22
3.4	Curvelet Transform Analysis	23
3.4.1	Curvelet Transform Output	24
3.4.2	Representation of image in curvelet domain	24
3.4.3	Sparsity	26
3.4.4	SSIM	27
3.4.5	Information content in phase	28
4	Sparse Feature Representation for Face Detection	29
4.0.6	Principal Component Analysis & Dimension Reduction	29
4.0.7	PCA in face detection	30
4.1	Stages in the development of the system	31
4.1.1	Training Phase	31
4.1.1.1	Selection of training data	32
4.1.1.2	Representation of image in curvelet domain	32
4.1.1.3	Training Algorithm	33
4.1.1.4	Classification procedure	35
4.1.2	Testing Phase	36
4.1.2.1	Confusion matrix & ROC	36
4.2	PCA directly on image intensity values	37
4.2.1	Performance	37
4.2.2	Effect of threshold	38
4.2.3	Effect of no. of eigen vectors	39
4.3	Observation	40
4.3.1	Eigen values of mean subtracted data	40
4.4	Performance Comparison	40
4.5	Graphical User Interface	41
5	Discussion and Future work	42
A	Background Mathematics	43
A.1	Covariance Matrix	43
A.2	Eigen vectors	43
	Bibliography	45

List of Figures

1.1	Face detection	1
1.2	Martin Cooper	1
1.3	Factors affecting detection(images from ORL[33] & Robotics Lab database[34])	4
2.1	Overview Of Face Detection System	10
2.2	Features	12
2.3	ROC	13
2.4	Matlab Output of cascaded object detector	14
3.1	Image and intensity values	15
3.2	Face image and edges	16
3.3	Two level wavelet decomposition	17
3.4	Block diagram for FRIT calculation	18
3.5	Curve aproximation using wavelet and curvelet	19
3.6	Curvelet in spatial domain	20
3.7	Curvelets and its Fourier Transform	21
3.8	Sub-bands in the frequency domain	22
3.9	Digital Curvelet Transform via wrapping	23
3.10	Wrapping of wedge into a rectangle	23
3.11	Curvelets coefficients at scales 1,2,3 and 4	25
3.12	Curvelets coefficients at scale 5 and 6	25
3.13	Variation in average sparsity with threshold	27
3.14	SSIM of the original and thresholded image with various threshold values	27
3.15	Variation in average SSIM with threshold in phase only reconstruction . .	28
4.1	First 10 eigen faces of the ORL Database of faces [33]	31
4.2	Faces from Yale B extended database[35]	32
4.3	Faces from ORL database of faces [33]	33
4.4	Flowchart	33
4.5	TPR vs Threshold	36
4.6	FPR vs Threshold	37
4.7	ROC	37
4.8	FPR vs threshold and TPR vs threshold	38
4.9	ROC	39
4.10	Variation of TPR with number of eigen vectors used	39
4.11	PCA implementation in Matlab	41
4.12	GUI of the system	41

List of Tables

1.1	Factors Affecting Detection	3
1.2	Applications Of Detection	4
1.3	Developments	5
2.1	Literature Survey	8
3.1	scale and Frequency sub-bands	24
3.2	Dimension of vector	26
4.1	Parameters for selecting the training data	32
4.2	Performance Comparison	40

Abbreviations

AUC	A rea U nder the C urve
CT	C urvelet T ransform
FFT	F ast F ourier T ransform
FNR	F alse N egative R ate
FPR	F alse P ositive R ate
MATLAB	MAT rix LAB oratory
PCA	P rincipal C omponent A nalysis
ROC	R eciever O perating C haracteristic
SSIM	S tructural S imilarity I ndex M easurement
TNR	T rue N egative R ate
TPR	T rue P ositive R ate

Dedicated to My Parents

Chapter 1

Introduction

Face detection is the process of locating faces in an image. Face detection can be considered as subset of object detection. Here the object to be detected is the face. Face detection is the process of locating face in a given image/video and face recognition is the process of identifying a person from an image. Face detection give the answer for the question, 'where is the face?' But in face recognition the question to be answered is, 'who is this?'. Face recognition is made possible by extracting the features of the face from the image. This can be done only when the face is detected. So face detection can be considered as the preliminary step in face recognition. What makes face detection different from normal object detection? Normal objects are rigid bodies with the same structure always. But in face detection, the object face is not perfectly rigid. It changes with changes in expressions of the face, appearance of the face etc. This makes the face detection a challenging task. The pictures shown below represent the output of face detection and face recognition system respectively.

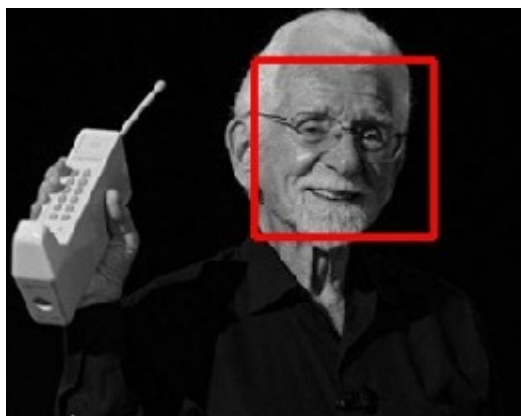


FIGURE 1.1: Face detection



FIGURE 1.2: Martin Cooper

1.1 Motivation & Objective

Face detection is one area in image processing which has undergone a wide research. There are many powerful algorithms available now. Hardware powerful enough to run those algorithms are also present now. Even though such powerful systems are available, today's face detection system works under certain constraints. The system performs well when operated under these conditions. For example the Viola Jones algorithm[1] is trained to detect only front facing images. It cannot detect faces oriented in any other direction. For detecting faces in any other orientation the system has to be trained with the corresponding type of training data. This will increase the computational burden of the algorithm. Similarly when the different parameters which affect the face detection process is considered it will in turn increase the computational time of the system.

Algorithms which work well in the laboratory, usually fails to have the same level of performance in real world scenarios. This is due to the fact that the conditions in the real world is different from restricted conditions under which the system is trained. In real world scenarios multiple parameters affect the system. A system which is trained to perform well for a specific factor fails when multiple factors are present. To have a good detection system it must be trained by considering all these parameters. But this will affect the computational time of the system. To reduce the computation time we have to reduce the number of computations. This is possible if the image can be represented with few non zero coefficients. One such method is to represent the image in a transform domain which provides a sparse representation for the image. Curvelet transform is efficient in representing images sparsely. It is efficient in representing curve like discontinuities. Since it provides a sparse representation it will result in less computations and reduced time complexity.

To further reduce the computations, a dimension reduction procedure can be applied. When the higher dimension data is represented in a lower dimensional space it will reduce the computations. One such dimension reduction method is the principal component analysis(PCA). The sparse representation of the image along with the dimension reduction procedure should effectively reduce the computation time.

The work done in this thesis can be summarized as follows.

- Analysis of existing face detection system

The analysis was mainly concentrated on the face detection algorithm developed by Viola & Jones[1]. Feature based classifier system was analyzed in detail.

- Analysis of Curvelet Transform

Fast Discrete Curvelet Transform via wrapping based algorithm was used for the computation of curvelet transform. Sparsity of the curvelet transform was analyzed and structural similarity index measurement (SSIM) parameter was used as the quality measurement tool.

- Development of a Face/Non-Face classifier.

A vector representation of the features is obtained from the curvelet transform. Further PCA is applied to reduce the dimension of the data. Performance of the classifier was measured using confusion matrix and receiver operating characteristic(ROC).

Performance comparison of classifier.

The performance of the above mentioned system is compared with a system which directly applies the PCA on the image intensity values.

1.2 Factors affecting detection

Some of the factors which affect the face detection are given below.

TABLE 1.1: Factors Affecting Detection

Factor	Example
Physical	Facial expression, aging, personal appearance (make-up, glasses, facial hair, disguise etc.)
Geometrical	orientation of face with respect to camera (full-frontal, profile ,oblique pose, in-plane rotation etc.),change in scale.
Imaging	Lighting conditions, camera variations, channel properties (when image is transmitted).

The important factors can be thought of as illumination, pose and occlusion. An example for the effect of these parameters on face images is shown below.

1.3 Applications of Face Detection

Some of the applications of face detection are given in the table.

One area of application is biometrics. But there are already some established methods like iris scan, fingerprint recognition and speech recognition. One advantage of the face recognition with respect to these methods is that, it doesn't require user consent.

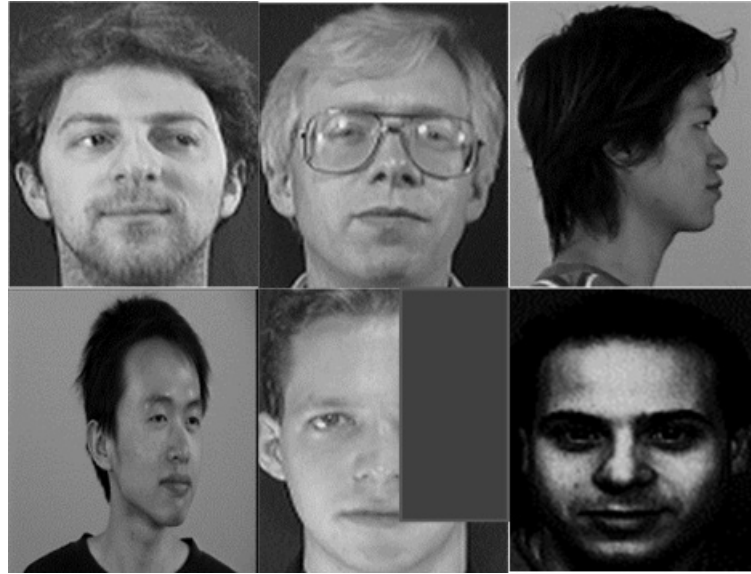


FIGURE 1.3: Factors affecting detection(images from ORL[33] & Robotics Lab database[34])

TABLE 1.2: Applications Of Detection

Area	Applications
Security	Access Control, User verification
Personal Security	Login information, Expression Interpretation
Biometrics	Person Identification, Automated identity verification
Law Enforcement	Video surveillance, suspect tracking
Artificial Intelligence	Computer Vision applications
Commercial	Video game systems, Camera centric applications

For all the other methods the user need to go near the system and interact with it for further processing. The above feature makes face recognition an excellent candidate for surveillance applications. Once the technology reaches the required performance level, it can be widely used for this application.

1.4 Developments in face detection

The first automated face detection system was developed by Kanade in the year 1973. Some of the important events in the history of face detection are given below

TABLE 1.3: Developments

Authors	Name	Year	Description
Takeo Kanade	Picture Processing System by Computer Complex and Recognition of Human Faces[24]	1973	First automated system. Operated on gray level images of 5-bits.Extracts feature points such as eyes,nose, mouth, chin etc. Trained to detect front facing images only. Not trained for profile face or occluded faces.
Sirovich & Kirby	Low dimensional procedure for characterization of human face[21].	1987	Based on Karhunen Loeve method. Representation of face using eigen pictures. Makes use of the symmetry of the face. Trained for only front facing images.
Turk & Pentland	Eigen Faces for Face Recognition[32]	1991	Eigen face based method. Time requirement of 400ms to complete detection. Performance degrades with illumination and orientation changes.Also performance decreases with scale(size of head).
Etemad & Chellapa	Discriminant Analysis for Recognition of Human Face Images[22]	1996	Based on Linear Discriminant Analysis. Gray scale images were used. Applicable only to front facing images.Slight variation in the illumination is acceptable.
Rowley, Baluja & Kanade	Neural-Network Based Face Detection [13]	1998	Works with gray-scale images. Detection Rate of 77.9% to 90%. Trained to detect front facing images only.Can be trained to detect oblique poses also.
Viola & Jones	Rapid Object Detection using a Cascade of Boosted Simple Features [1]	2001	Works with gray-scale images. Trained to detect only font facing images. Fastest algorithm to detect frontal faces.
Naruniec & Skarbek	Face Detection using discrete Gabor Jets [3]	2007	Based on the Gabor Transform. Works by detecting the fiducial points from the edge information. Computational complexity is higher than Viola-Jones method.

1.5 Organization of the thesis

The thesis consists of five chapters as follows

Chapter 1: Introduction. This chapter gives the objective of the thesis and the motivation behind the selection of this work.Some of the important works in the history of face detection is also mentioned in the chapter

Chapter 2: Feature Based Face Detection. This chapter has literature review on some of the existing methods in the area of face detection, which were analyzed as a part this work. It also gives an overview of a feature based face detection system. It explains the basic blocks in a face detection system and the performance measure which can be used to compare different algorithms. The discussions in this chapter is mostly based on the work done by viola & Jones.

Chapter 3: Face Image in Curvelet Domain. This chapter explains what is a curvelet, how curvelet transform is calculated and how exactly it will be useful in the face detection process. It gives an overview about the implementation of Fast Discrete Curvelet Transform(FDCT). It also analyzes the representation of face images in curvelet domain. The sparsity of the transform is analyzed with the help of structural similarity index measure(SSIM).

Chapter 4: Sparse Feature Representation for Face Detection. This chapter explains the feature representation in the curvelet domain and the dimension reduction procedure using PCA. It discusses about the implementation algorithm. The performance of system using curvelet transform is compared with the system which works directly on the intensity values.

Chapter 5: Discussion and Future work. This chapter summarizes the work done in this thesis. It also discusses the work which can be implemented in the future.

Chapter 2

Feature Based Face Detection

This chapter gives an overview of the literature survey work that has been completed as a part of this work and an overview of feature based face detection system.

2.1 Literature Survey

One of the important method in the field of face detection is the Viola Jones method [1]. This system was trained to detect only front facing images. So the system worked well for front facing images and it is the fastest algorithm in its category. No color information was used in this algorithm. The detection rate was mentioned to be equal to that of the Rowley-Baluja-Kanade detector [13]. The main drawback of this method was that it was trained to detect only frontal faces from images. This algorithm can be trained to detect faces having different orientations. For detecting faces in other orientations additional training is required. The paper by Jones *et al.* [9] is an extension to the paper published by Viola *et al.* [1]. This method was trained to detect rotated faces and profile faces. A decision tree approach was used to determine the viewpoint from the image. Twelve rotation classes each of angle 60° ($12 \times 60 = 360$) was used for the detector.

In the paper, by Lienhart *et al.* [12] rotated Haar-like features are used. The basic set is extended by a set of 45° rotated features. This algorithm was able to reduce the false detection rates. The speed of execution was less when compared with the Viola-Jones [1] frontal face detection method.

The paper by Gupta *et al.* [7] studies the effect of various parameters (maximum angle and maximum deviation) on the classifiers for face detection. It was found that the best performance for a classifier is achieved at moderate values of the maximum angle and maximum deviation.

TABLE 2.1: Literature Survey

Paper	Remarks
Rapid Object Detection using a Cascade of Boosted Simple Features [1]	Works with gray scale images. Trained to detect only front facing images. Fastest in this category. Cannot detect profile and in-plane rotated faces.
Fast multi-view face detection [9]	Trained to detect profile and rotated faces. Detection rate was less and the time requirement was more compared to Viola-Jones method [1].
An Extended Set of Haar-like Features for Rapid Object Detection [12]	Trained to detect profile faces. Uses gray-scale image. Detection rate was less compared to Viola-Jones [1] method.
Analysis of Training parameters for classifiers based on Haar-like features to detect human faces [7]	Classifiers were trained to detect rotated and illumination varied faces. Performance of the classifiers were studied under different conditions(maximum angle and maximum deviation). Detection rate is reduced compared as only few features were used. Best performance for a classifier was achieved at moderate values of the parameters.
A short Introduction to boosting [2]	A generic algorithm to improve the performance of an existing learning algorithm
Face Detection using Discrete Gabor Jets [3]	Face detection based on edge related information. Computational time requirement is more. Extracted information for face detection can also be used for face recognition.

From the literature it was clear that even though fast algorithms are available for frontal face detection, the detection rate or speed was affected when the same was trained to detect tilted faces, profile faces and at different illuminations. In order to have a good face detection system it should detect the faces under these various conditions(illumination ,pose and occlusion) without sacrificing on the time complexity.

2.2 Feature based detection

Most of the discussions in this chapter are based on Viola-Jones [1] face detection paper. This can be used as a platform for the development of a face detection system.

2.3 Different methods of detecting a face

When an image is given to a human being he can identify the objects present in the image. He can identify a face and he can differentiate the face from other objects present in the image and the background. But for a machine all the images are same. For a machine all the images are nothing but a two dimensional array of integer values (0 to

255 for 8-bit gray-scale image). So a machine cannot differentiate objects in an image as we humans do. So it is up to us to develop a method to identify the faces from these integer values.

Some of the methods which are used for face detection are given below.

- Feature
- Skin Tone(In color images)
- Shape
- Motion(In videos)
- Combination of the above methods

2.3.1 Overview of face detection system

Any artificially intelligent system has got two stages in the development. First is the training phase and the second is the testing phase. It is during the training phase that we train the system so as to carry out its objective. Once the training is over the real data is given to the system and its output is verified.

How well the system performs depends on how well you have trained the system. This training stage includes the collection of training data, implementing algorithms etc. Better algorithms will result in better performance.

Face detection system can be considered as an artificially intelligent system which is trained to detect faces from an image. It has also got the two stages in development as shown below.

- Training Phase
 1. Training Data
 2. Feature Selection
 3. Define Classifier
- Testing Phase
 1. Slide Window
 2. Score by the Classifier
 3. Mark the face if it is present

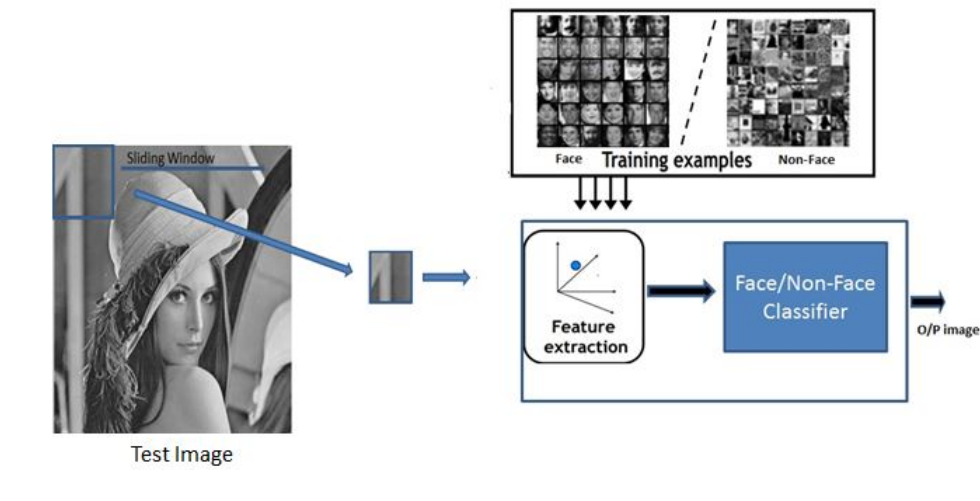


FIGURE 2.1: Overview Of Face Detection System

2.3.2 Training Phase

During this phase we have to get the training data. We have to define a model for the face in the corresponding domain (It can be spatial domain or any other transform domain) and an algorithm for detecting faces in that domain. Once these prerequisites are completed the training can be done. Some of the steps in the training are explained below.

2.3.2.1 Obtaining training data

As face detection system tells us about the location of a face in an image, two kinds of images are possible. They are face images (image of face) and non-face images (image which doesn't contain any face). The training data should have some variations (illumination level, age of person, gender, region, pose etc.). This will help to improve the detection capability of the system. The training data will be of the same size as the size of the basic window, which will be used in the testing phase. Basic window size represents the smallest face that can be detected from an image.

The face/non-face images can be obtained online. The face images can be obtained from online face databases, and the non-faces can be obtained by a random search in internet, all scaled to the basic size. Some of the face databases available are mentioned below

- ORL Database (The database of faces)

The data set consists of different images of 40 subjects taken under varying illumination, expression and occlusion.

- CMU PIE Database(Pose illumination and expression database)

The data set consists of 41,368 images of 68 people. The images were taken under different facial expressions, poses and illumination levels.

- FERET Database

The data set consists of face images taken under different facial expression, pose and illumination.

- LFW Database(Labeled faces in the wild)

The data set contains more than 13,000 images of faces collected from the web.

- Yale Face Database

165 grayscale images in gif format of 15 individuals. There are 11 images per subject.

- Yale Face Database B

16128 images of 28 human subjects under 9 pose and 64 illumination conditions.

- The AR Face Database

The data set consists of over 4000 color images of 126 people under different facial expression, pose and occlusions.

2.3.2.2 Define features

Feature defines the face. Feature differentiates face from other objects and background in the image. So the first step in the development of detection algorithm is to define the features. The features can be defined in spatial domain or in any transform domains. Feature can be some pattern which characterizes the face(in spatial domain) as used in the Viola-Jones [1] method. The better the detection power of the feature, better the algorithm will be.

Examples for features used in the Viola-Jones [1] algorithm is given in figure 2.2. These features are called Haar-like features because of the similarity with Haar wavelets.

The first feature is based on the property that the region of the eyes are always darker than region of the cheeks. The second feature is based on the property that the region of eyes are darker than the bridge of the nose.

In some cases a reference graph which represents the fiducial points(eye corners, mouth corners, nose corners) as seen in the paper Face Detection using discrete Gabor Jets by Narune et al. [3] can be used as the pattern to detect the face.

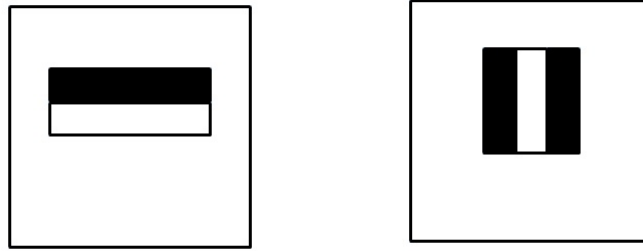


FIGURE 2.2: Features

2.3.2.3 Define Classifier

A single feature cannot detect all the faces. Chances are high that it might detect non-faces also. This happens due to the fact that there might be some regions in the image which follow the same pattern of the feature. So in order to detect faces alone we will have to define multiple features. A set of these features can be combined to form a classifier. In face detection the classifier used is a binary one i.e. the output can have only two possible states. The output of the classifier is either 'face is present' (state 1) or 'face is not present' (state 2). So classifier can be considered as the basic block in the detection process. A series of classifiers can be cascaded to form the overall detector

2.3.3 Testing Phase

During the testing phase an image of any random size is given as the input to the system. A window of basic size is defined and this is used to scan the entire image starting from the top left corner to the bottom right corner. Once the window selects a portion of the image the features of the classifier are applied to it to detect the face. This process is done for all the windows selected from the image. Scaled version of the window are then used to detect a face at a different scale. Once the face is detected it will be marked using a rectangular window.

2.4 Possible errors in face detection

There are mainly two kinds of error that can happen in a face detection system.

- Not detecting a face which is present in an image.
- Detecting a face in an image where actually no face is present.

2.5 Performance comparison

Performance of a face detection system can be obtained from the Receiver Operating Characteristics(ROC). Before calculating ROC one need to find the confusion matrix for the problem. Confusion matrix for face detection problem is given below.

$$\text{Confusion Matrix} = \begin{bmatrix} TPR & FNR \\ FPR & TNR \end{bmatrix}$$

- True Positive Rate(TPR)

$$TPR = \frac{\text{True positives}}{\text{Total positives}} = \frac{\text{No. of correctly classified faces in the dataset}}{\text{Total no. of faces in the dataset}}$$

- False Negative Rate(FNR)

$$FNR = \frac{\text{False negatives}}{\text{Total positives}} = \frac{\text{No. of faces wrongly classified as nonfaces}}{\text{Total no. of faces in the dataset}}$$

- True Negative Rate(TNR)

$$TNR = \frac{\text{True negatives}}{\text{Total negatives}} = \frac{\text{No. of correctly classified non - faces}}{\text{Total no. of nonfaces in the dataset}}$$

- False Positive Rate(FPR)

$$FPR = \frac{\text{False positives}}{\text{Total negatives}} = \frac{\text{No. of nonfaces wrongly classified as face}}{\text{Total no. of non - faces in the dataset}}$$

ROC is a graph of True Positive Rate(TPR) vs. False Positive Rate(FPR).

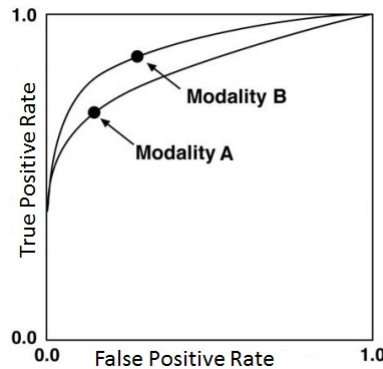


FIGURE 2.3: ROC

The integral of the curve produces the Area Under the Curve(AUC).This indicates the accuracy of the classifier. For ideal detector false positive rate is zero and true positive rate is one. Such system will have a unity AUC. ROC can be use to compare performance of different algorithms. In the above graph modality B has better performance than modality A. For a given FPR modality gives better TPR. Another performance parameter is the speed of execution of the detector.The time required for detection should be within a limit(so that it can be used for real time applications).

2.6 Matlab output for Cascaded Object Detector

The Viola-Jones[1] detector implementation is available in MATLAB. This detector was used to create a program to detect the frontal faces. The result obtained is shown below.

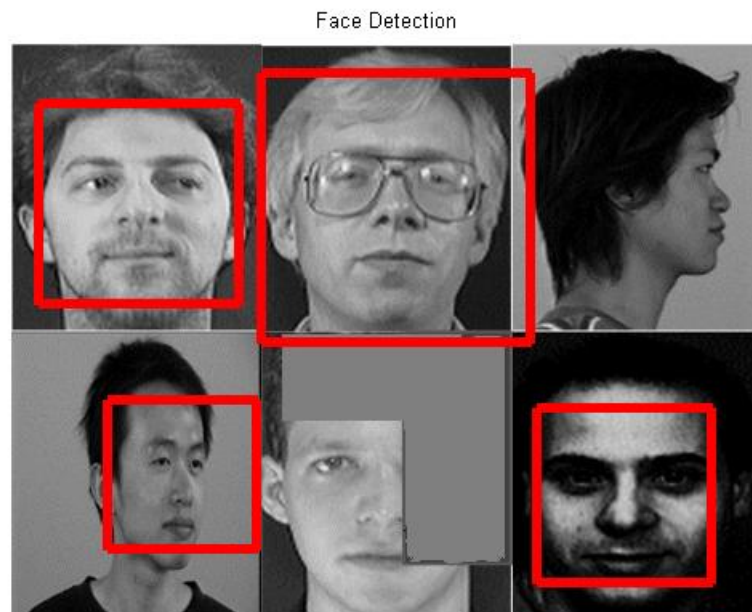


FIGURE 2.4: Matlab Output of cascaded object detector

Chapter 3

Face Image in Curvelet Domain

In order to have a good face detection system, it should incorporate the variables like pose, illumination and occlusion into the system. But this should not increase the computational time required for the detector. One way of achieving this is to represent the image sparsely. This sparse representation will have less non zero values and so the number of computations will be reduced.

Any image in the spatial domain can be considered as a set of pixels. Each of these pixels will correspond to some specific intensity values. In most of the images all of the pixels will have non zero intensity values. An example is below.



FIGURE 3.1: Image and intensity values

Many of the existing face detection systems works in the spatial domain. Template matching techniques are used to detect the face from the image. When computations are done in the spatial domain all the non-zero intensity values have to be taken into consideration for the computations. Since the number of non-zero intensity values is large, the number of computations required will be more. This will make the algorithm slower.

To increase the speed of the algorithm, the number of computations should be reduced. For reducing the computations the number of non-zero values required

to represent the image should be reduced. If the number of nonzero values is less then, while doing the computations only these non-zero coefficients need to be considered. This will reduce the number of computations which in-turn increases the speed of the algorithm. So the key for speed improvement lies in the sparse representation of the image.

3.1 Sparse signal

The literal meaning of the word sparse is something which is small in quantity and which is spread over a wide area. In signal processing field if a signal has few non-zero coefficients then it can be considered as sparse. To obtain a sparse representation we have to represent the signal in a basis/dictionary which can characterize the image with a few non-zero coefficients. For this the input image must be transformed from the spatial domain to the transform domain.

3.2 Selection of a transform

There are so many transformations available and which one should be used to get the sparse representation. The selection of a particular transform depends on the type of input used in the application. In face detection application the image under consideration is a face image. If we analyze the image we can see that the image consists of regions of constant intensity as well as edges as in the case of any images. In order to get the edge information in the image an edge detection operation can be done on the face image. An example is shown below. From the

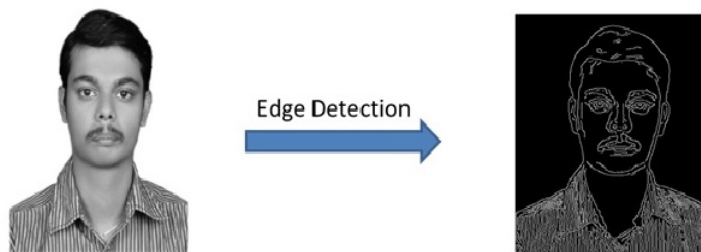


FIGURE 3.2: Face image and edges

edge detected diagram it is visible that the edges in a face image are not straight line edges but curved edges. Similarly for any general image the edges may not be a straight line. So the transform which is selected should efficiently represent the curve discontinuities in less number of coefficients. Some of the available transforms are Wavelet Transform, Ridgelet Transform and Curvelet Transform.

3.2.1 Wavelet Transform

Wavelet Transform is a multi-resolution tool used in signal processing. In two dimensional wavelet transform, the basis functions have got three directions namely horizontal, vertical and diagonal. Wavelet can represent edges in these directions with the help of these basis functions. Edges in any other direction or curves have to be approximated with the available basis functions. This indicates that number coefficients required to represent curve-like discontinuities and arbitrary edges are more. So wavelet transform cannot provide sparse representation for images with curved edges. This clearly indicates that wavelet transform cannot provide sparse representation for curve-like discontinuities.

A two level decomposition of images containing edges and curves is done using wavelet transform. The output obtained is shown below.

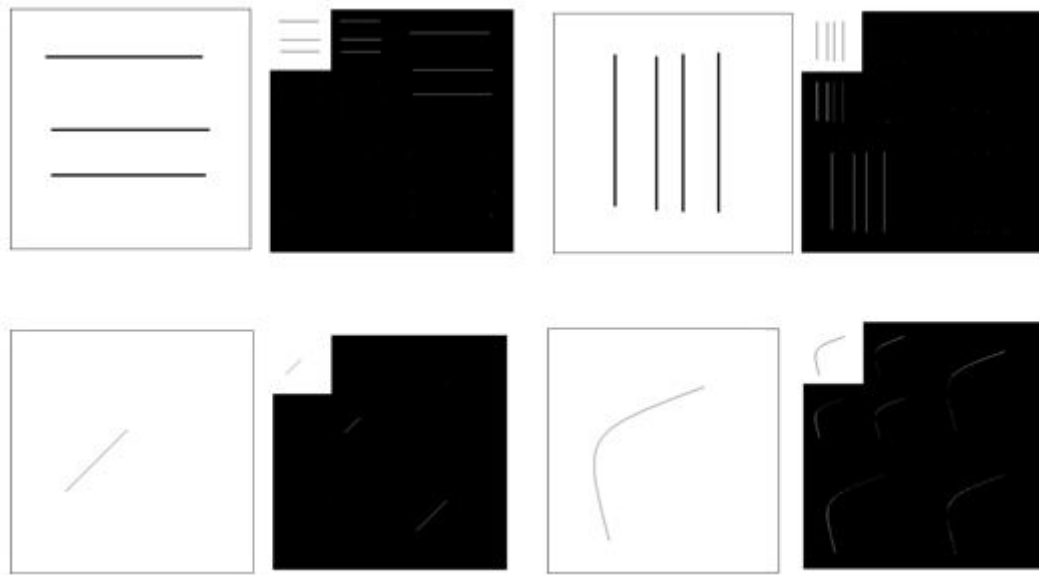


FIGURE 3.3: Two level wavelet decomposition

First three decomposition's indicates that wavelet transform can represent the edges in horizontal, vertical as well as diagonal direction using few coefficients. But when there is a curved edge in the input image the number of coefficients required to represent the curve is more. The curved edge is approximated in terms of the horizontal, vertical and diagonal edges. This indicates that wavelets require more coefficients in representing curves and edges in arbitrary direction.

Classical Discrete wavelet transform (DWT) does not provide good directional selectivity (three directions). The directional selectivity can be improved by using complex wavelets (six directions). The approximation of curves, provided by the

complex wavelets is better than that of the classical wavelets. Still this is not efficient in sparsely representing curves.

3.2.2 Ridgelet and Curvelet Transform

An anisotropic geometrical wavelet transform was proposed by Candes & Donoho in the year 1999. It is named as the Ridgelet transform. The parameters associated with the Ridgelet transform are

1. Scale
2. Translation
3. Rotation

Because of the presence of the additional rotation parameter it is considered as a geometrical wavelet i.e. a wavelet which can be rotated in any direction. Wavelet can represent point discontinuities and ridgelets can represent line continuities. Ridgelet transform can represent global-straight line singularities. But global straight line singularities are rarely present in practical images. So ridgelet transform cannot provide sparse representation for curve-like discontinuities. The block diagram for the calculation of Finite Ridgelet Transform(FRIT) is given below.

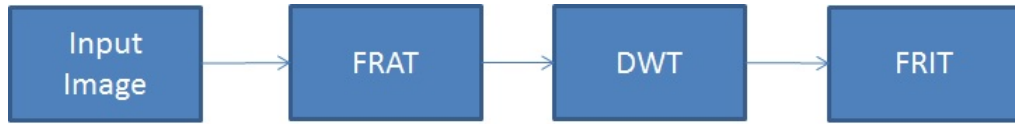


FIGURE 3.4: Block diagram for FRIT calculation

FRIT is calculated in two steps : a calculation of finite radon transform(FRAT) and an application of wavelet transform. FRAT is calculated in two steps:

1. Calculate the two dimensional Fast Fourier Transform(FFT) of the image. Obtain 32 radial directions from this. Apply inverse FFT on these.
2. One dimensional wavelet transform is applied to these radial directions.

Ridgelet transform cannot represent local straight line singularities. To mitigate this problem a block ridgelet based transform was proposed by Candes and Donoho in the year 2000. This is called as the Curvelet transform. The idea behind this is to divide the image into different blocks and apply the Ridgelet transform separately in each block. This implementation of Curvelet transform is called as the first generation curvelet transform(also known as curvelet99). A much simpler version called Fast Discrete Curvelet Transform(FDCT) was proposed by Candes and Donoho in the year 2005.

3.3 Fast Discrete Curvelet Transform (FDCT)

This is also known as the second generation curvelet transform. In this a frequency partition technique was used to calculate the curvelet transform. Discrete curvelet transform can sparsely represent curve-like edges. What makes it possible for the curvelet transform to provide a sparse representation for curve like edges? It comes from the scaling property. Curvelets follow anisotropic scaling whereas wavelets follow isotropic scaling. In isotropic scaling an object is scaled with same constant value along all the axes. In anisotropic scaling the constant of scaling will be different along different axes. In isotropic scaling the shape of the object remains same as the scale is changed, but in anisotropic scaling the shape changes as the scale changes. curvelets obey parabolic scaling law i.e. $\text{width}=\text{length}^2$

An approximation of a curve using wavelets as well as curvelets is given below.

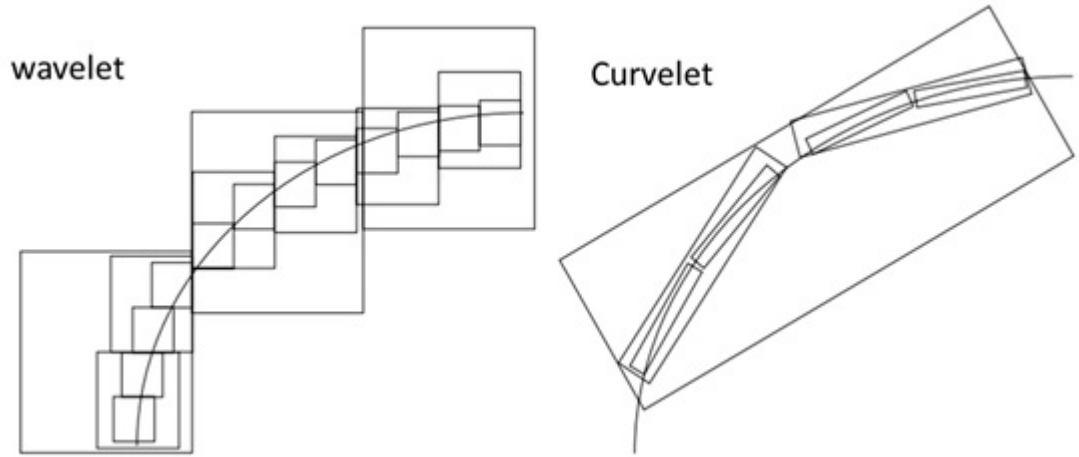


FIGURE 3.5: Curve approximation using wavelet and curvelet

From the diagram it is clearly visible that as the curvelet is shrunk it closely follows the curve. This is not possible in the case of wavelets as they follow isotropic scaling. This shows that curvelets require only few coefficients to represent a curve and can provide sparse representation for curve-like discontinuities.

How much efficient is the Curvelet transform in sparsely representing curve-like discontinuities? The error obtained in using a curvelet approximation for a signal is mentioned as $\|f - f_m^C\|_{L_2}^2 = O((\log m)^3 m^{-2})$ [16], where m is the number of non-zero coefficients.

3.3.1 Curvelet

The spatial domain representation of a Curvelet is shown below. This diagram shows a Curvelet at a particular scale, orientation and location. To be specific this



FIGURE 3.6: Curvelet in spatial domain

curvelet represents an edge in the diagonal direction. The frequency is varying in the off diagonal direction. The support for the curvelet is elliptical in shape. By varying the three parameters the basis functions of the curvelet transform can be obtained. Curvelets are needle shaped elements at finer scale and at the coarsest level the curvelet loses its directionality.

3.3.1.1 Curvelet categories

Curvelets can be considered to belong to one of the following three categories.

1. A curvelet which does not intersect a discontinuity. The magnitude of the curvelet coefficients will be zero.
2. A curvelet which intersect partially with a discontinuity. The magnitude of the curvelet coefficients will be near to zero.
3. A curvelet which does intersect length-wise with a discontinuity. The magnitude of the curvelet coefficients will be much greater than zero.

3.3.2 Digital Curvelet Transform

The curvelet transform coefficients are defined by

$$C^D(j, l, k) = \sum_{0 \leq t_1, t_2 < n} f[t_1, t_2] \overline{\phi_{j,j,k}^D[t_1, t_2]} \quad (3.1)$$

The curvelet waveform is not used in the implementation of the curvelet transform. In digital curvelet transform most of the computations are done in the frequency domain. In order to understand curvelet transform, curvelets should be analyzed

in the frequency domain.

In Fast Discrete Curvelet Transform there are two methods by which we can calculate the curvelet transform. They are

1. Digital Curvelet Transform via Unequally Spaced Fast Fourier Transform(USFFT)
2. Digital Curvelet Transform via Wrapping Function

The Wrapping based method is faster than USFFT based method[16] and the same is used in this work. As said earlier the Curvelet waveforms are not used in the implementation of the Curvelet Transform. They are introduced to provide an easy understanding of Curvelet transform. The fourier transform of curvelets are calculated using FFT algorithm. Curvelets and their FFT outputs are shown in figure 3.7. It can be seen from the diagrams that the frequency band corresponding to a curvelet is normal to the edge of curvelet or along the direction of change in frequency

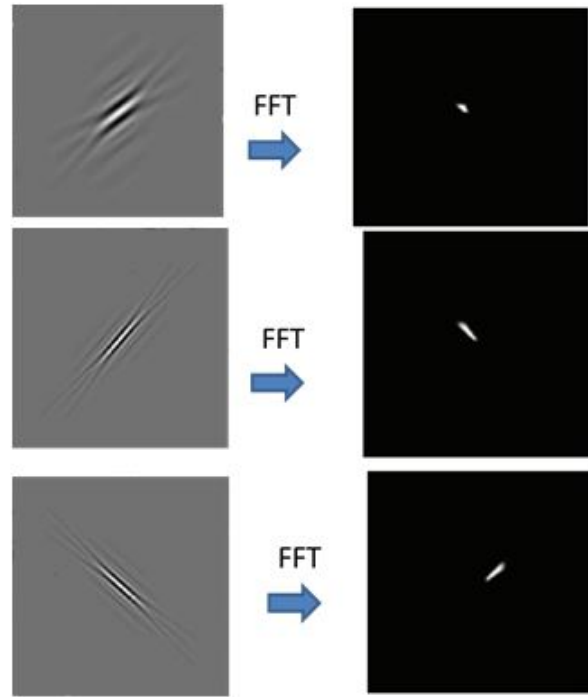


FIGURE 3.7: Curvelets and its Fourier Transform

The fourier transforms are calculated using the FFT method. The output of the fourier transform indicates that a curvelet at a particular scale and direction represents a localized region in frequency i.e. it corresponds to particular sub-band in the frequency plane. Each of these sub bands has a wedge shaped support. So the entire frequency plane can be divided into different wedge shaped frequency sub-bands which correspond to curvelets at a particular scale and orientation. This

tells us that instead of using the curvelets directly to calculate the coefficients, the frequency plane of an image can be divided into sub bands and an inverse FFT operation is done on each of these to get the curvelet coefficients.

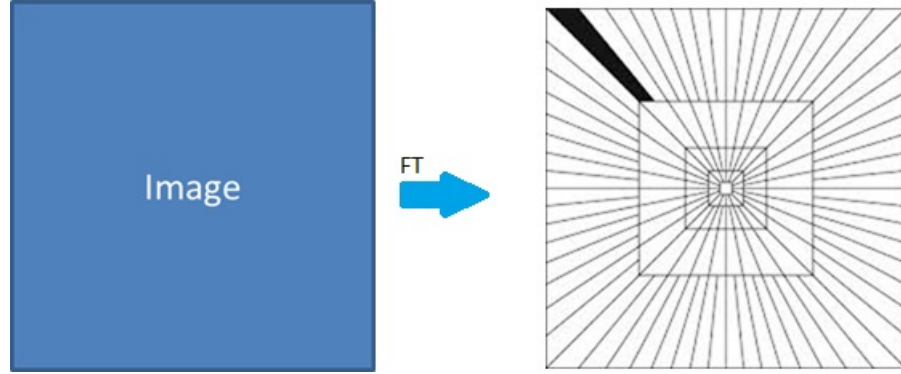


FIGURE 3.8: Sub-bands in the frequency domain

3.3.3 Digital Curvelet Transform via wrapping

The various steps involved in the calculation of FDCT via wrapping is given below.

1. Take 2D FFT of the input image.
2. Partition the frequency plane into different wedge shaped regions corresponding to curvelets at a scale and direction.
3. Wrap each of the wedges into rectangular region around the origin.
4. Apply 2D inverse FFT to each of the wrapped window to obtain the curvelet coefficients.

The steps involved in the calculation of curvelet transform coefficients corresponding to a single curvelet at a scale and orientation is shown below.

3.3.3.1 Wrapping

In order to get the curvelet coefficients an inverse FFT should be done on the wedge shaped frequency sub-band in the frequency domain. Since the sub-band is having a wedge shape inverse FFT cannot be applied to it directly. The wedge should be converted into a rectangle before applying the inverse FFT. Wrapping is the process of converting the wedge shaped region in the frequency plane to rectangular region. The various steps involved are given below.

1. Tile the wedge in the frequency plane. Now the support has the shape of a parallelogram.

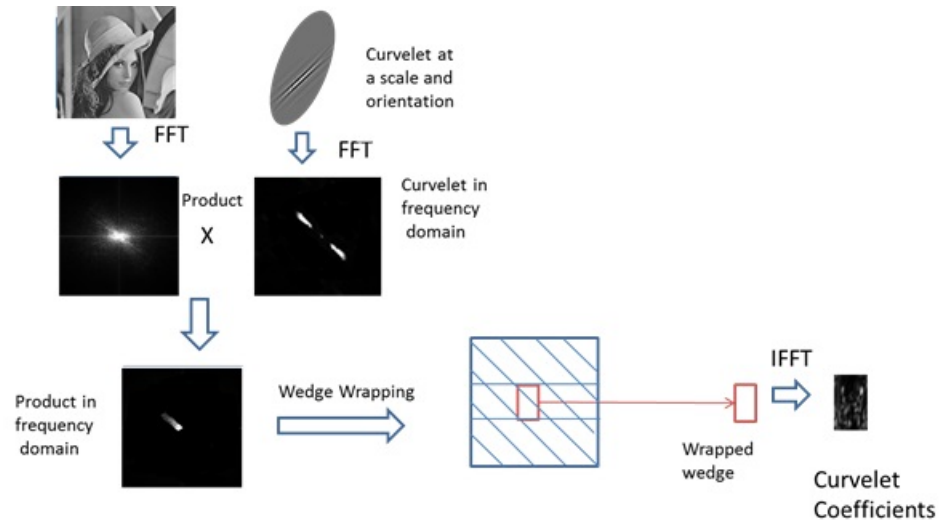


FIGURE 3.9: Digital Curvelet Transform via wrapping

2. Define a rectangular window at the origin which has the same dimension as that of the parallelogram.
3. This rectangular window now has the entire information from the wedge wrapped inside it

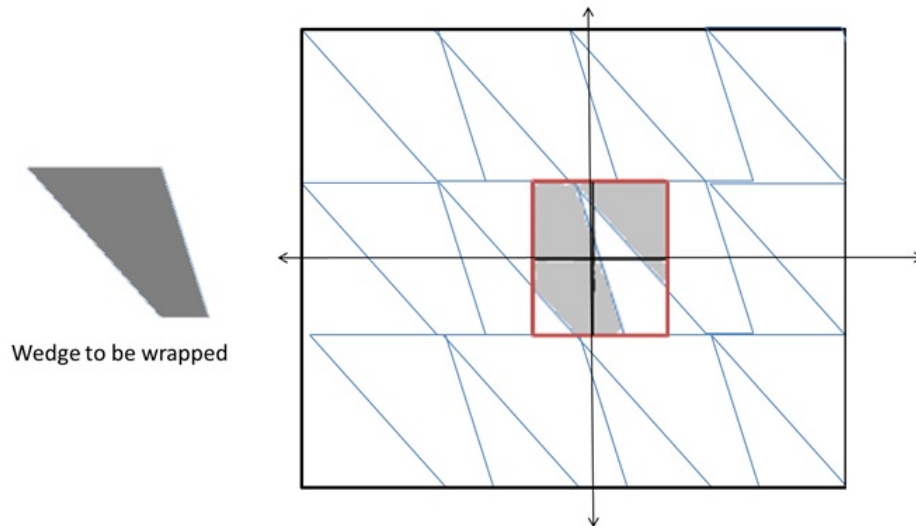


FIGURE 3.10: Wrapping of wedge into a rectangle

3.4 Curvelet Transform Analysis

Curvelet transform provides efficient representation of curve like discontinuities. Images consists of curves. So curvelet transform should be able to provide an

efficient and sparse representation for the signal. Detailed analysis of the curvelet transform is done in the following sections.

3.4.1 Curvelet Transform Output

The Curvelet coefficients obtained for Lena Image is shown. The different parameters used are mentioned below.

Curvelab Version : CurveLab 2.1.3

Website : www.curvelet.org

Matlab file : fdct _ wrapping.m

Input image(X) : Lena.jpg Grayscale with a dimension 512x512

is _ real : 0 Complex Curvelet Coefficients

finest : 2 Wavelet at the finest level

nbscales : 6 No. of levels of decomposition

nbangles _ coarse : 8 No. of angles at the second coarsest level

TABLE 3.1: scale and Frequency sub-bands

Scale	No of sub-bands(orientations)
1	1
2	8
3	16
4	16
5	32
5	1

imshow function is used to plot the absolute value of the complex coefficients.

3.4.2 Representation of image in curvelet domain

Curvelet transform is obtained by taking the FFT of the given image and then taking the IFFT of the frequency partitions. The FFT coefficients on the upper and lower half are complex conjugates to each other. So for representing an image instead of using the entire coefficients, one half of the coefficients can be used. This helps in reducing the dimension of the representation. The columns in each of the subbands are appended to form a single column vector representation of the image. For an image of dimension 192x168 the length of the vector representation for different number scales is given below.

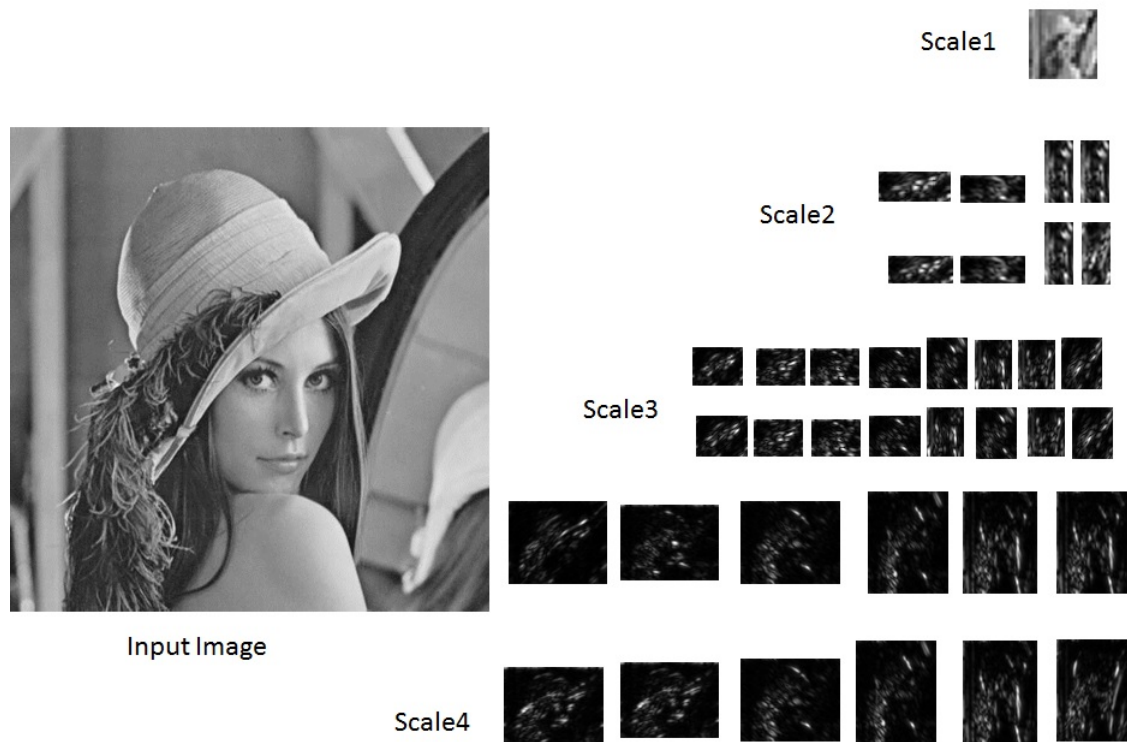


FIGURE 3.11: Curvelets coefficients at scales 1,2,3 and 4

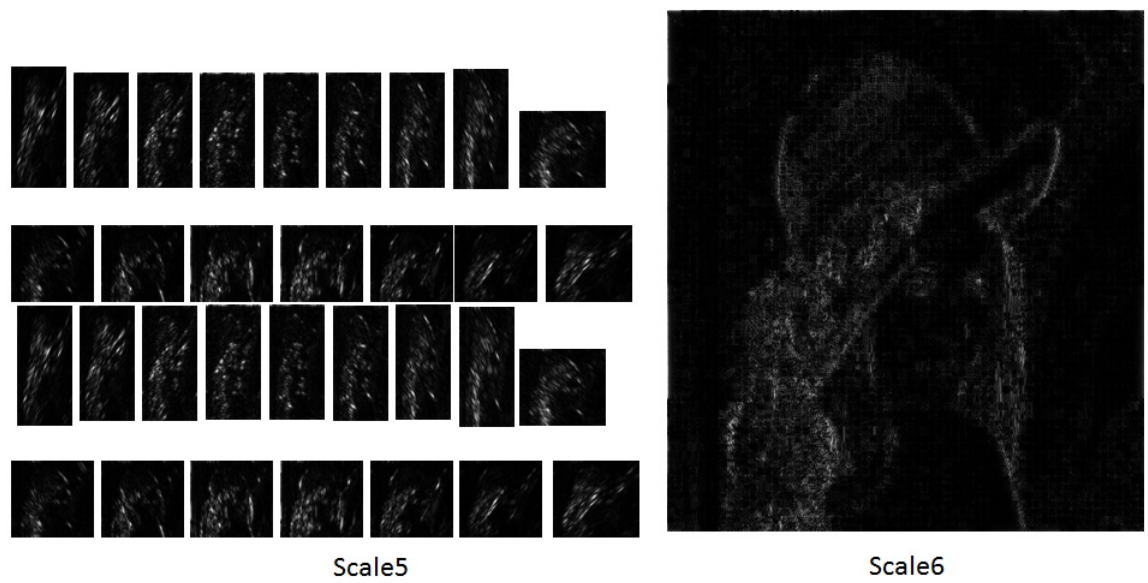


FIGURE 3.12: Curvelets coefficients at scale 5 and 6

TABLE 3.2: Dimension of vector

Image Dimension	No. of scales	Dimension
192x168	2	46833
192x168	3	61743
192x168	4	62723
192x168	5	63055

3.4.3 Sparsity

Sparsity corresponds to the number of non-zero values in a signal. Suppose a signal of dimension N has got k non zero values then the signal is said to be k -sparse.

$$\text{signal } s = [k \text{ nonzeros and } (N - k) \text{ non zero values}]_{Nx1}^T$$

Real signals may not be always sparse. But they can have a sparse form in another representation. For example an image in spatial domain can have sparse representation in fourier, wavelet or curvelet transform. This is obtained by approximating the small coefficients to zero in the transform representations. This sparse representation helps in compressing the signal as only few non zero coefficients are required to represent signal. The different training images are selected from Yale B database. The advantage of Yale B image is that the face images are actually cropped to have only the face with no background. This helps in the proper representation of the face images. Curvelet transform of the image is computed and the coefficient with the maximum value in each of the sub bands is identified. In the next step the coefficients whose value is less than the mentioned percentage of the maximum value are discarded. This might result in the loss of structural information. But how much information is lost? To measure this we need to use a parameter for evaluation. This is measured by taking the SSIM (structural similarity index measurement) value between the reconstructed image and the original image. The variation of the sparsity of the image after threshold is given in fig 3.13. The plot indicates that the number of non-zero coefficients increases as the threshold value is increased.

Choosing a specific threshold value depends on the amount of information retained in the curvelet coefficients after threshold. For measuring the structural similarity SSIM value is used. The variation of SSIM with the threshold value is given in the figure 3.14.

From the graph it is clear that as the threshold value is increased from $\frac{\text{max value}}{5}$ the SSIM value almost remains constant. Below which the SSIM value decreases drastically as the threshold value is increased. An effective value of threshold should

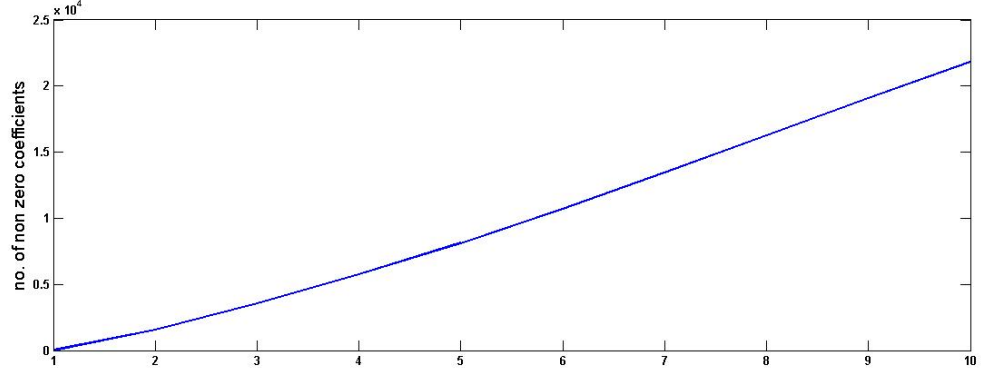


FIGURE 3.13: Variation in average sparsity with threshold

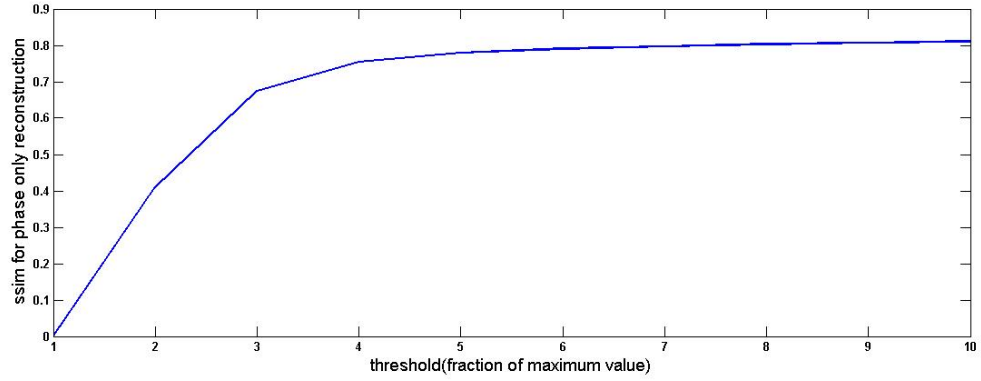


FIGURE 3.14: SSIM of the original and thresholded image with various threshold values

give reasonably high value of SSIM and low value of sparsity. Both these parameters must be considered while deciding a threshold.

3.4.4 SSIM

SSIM stands for structural similarity measurement index. It is a quality assessment tool. It tell how much structurally similar two images are. Higher the ssim value more structurally similar are the two images. This can be used to measure the quality of a reconstructed image with the original image. The equation governing ssim of two signals x and y is given by

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \quad (3.2)$$

where , μ_x is the mean of signal x, μ_y is the mean of signal y, σ_x is the standard deviation of signal x, σ_y is the standard deviation of signal y, σ_{xy} is the variation of x and y together and C_1 and C_2 are constants,

3.4.5 Information content in phase

As in the case fourier transform and wavelet transform the phase retains the edge information of the signal. For analyzing that phase only reconstruction of image is done after taking the curvelet transform. SSIM of the original image and the reconstructed image is computed. The process is done on a dataset of 37 face images obtained from Yale B database. The values obtained values are min value=0.8079, max value=0.8797 and mean=0.8427. These values indicate that the phase indeed retains most of the information content in the signal. The variation in SSIM with phase only reconstruction for different values of threshold is given in figure 3.15.

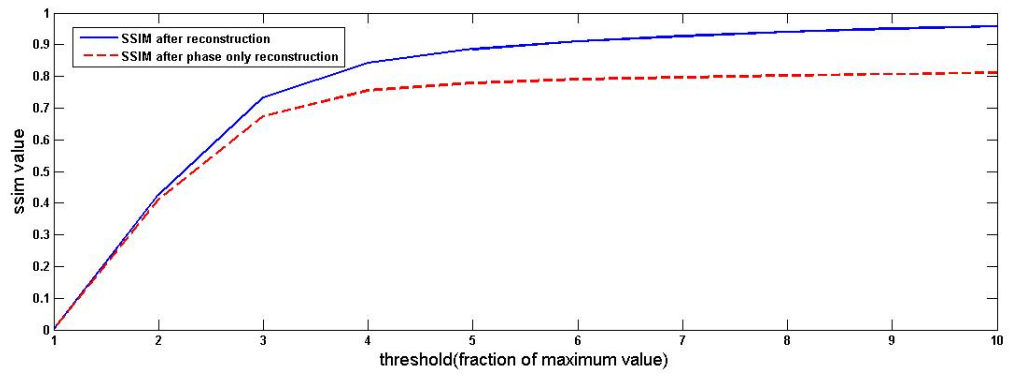


FIGURE 3.15: Variation in average SSIM with threshold in phase only reconstruction

Chapter 4

Sparse Feature Representation for Face Detection

The representation of image in curvelet transform provides a sparse representation. This sparse representation reduces the computation time. Further reduction in computation time can be achieved by applying a dimension reduction procedure. The output of a curvelet transform consists of higher dimensional data. This higher dimensional data can be projected into a PCA subspace for a lower dimensional representation.

4.0.6 Principal Component Analysis & Dimension Reduction

PCA is a method used for reducing the dimension of data. It achieves dimension reduction by removing the redundancy in the data representation. The principal components are the directions along which the data has got maximum spread(maximum variance). The principal components are obtained by calculating the eigen vectors of the covariance matrix. These eigen vectors forms the basis vectors for the PCA space. The eigen space will have its origin at the mean of the original data. Eigen values indicate the amount of spread in that particular eigen direction. If the eigen value is very small that indicates very small spread of data in that direction. So one can neglect those directions which does not contribute much to the data. Removing those eigen vectors will not affect the data representation much as it has got very less contribution. Thus by removing all such unnecessary eigen directions, we can achieve a reduction in the dimension. Any higher dimensional data can be projected into the PCA subspace to get a lower dimensional representation. Only few of the eigen vectors are required for representing the data efficiently. Rest of the eigen vectors doesn't contribute much in representing the data. This is shown in the figure [4.10](#).

$$input\ data = \sum_{i=1}^k w_i E_i \quad (4.1)$$

where w_i is the weight of the i^{th} eigen vector, k is the number of eigen vectors selected and E_i is the i^{th} eigen vector. The new dimension k will be less than the original dimension N of the data. So when an N -dimensional vector is projected into the k -dimensional space it will result in a k -dimensional representation of the original data.

4.0.7 PCA in face detection

The training images are converted into column vectors and are put in the column of matrix A .

$$B = [\Gamma_1 \Gamma_2 \dots \Gamma_L] \quad (4.2)$$

where Γ_i corresponds to the vector representation of the training data and L is the total number of training images. The mean vector is calculated by taking the mean along the rows. The mean vector Ψ is given as

$$\Psi = \begin{pmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \\ \vdots \\ \mu_{MN} \end{pmatrix}_{MN \times 1}$$

MN is the dimension of the vector representation of image in the curvelet domain. The mean subtracted data is defined as $\Phi_i = \Gamma_i - \Psi$. The matrix A is defined as

$$A = [\Phi_1 \Phi_2 \dots \Phi_L] \quad (4.3)$$

The covariance matrix is given by

$$C = AA^T \quad (4.4)$$

The eigen values of the covariance matrix is computed. In the case of face detection these eigen vectors are called as the eigen faces. Convert the eigen vector back to an image which has the same dimension as the training data to get the eigen face. When this matrix is displayed it shows a face image. Once eigen faces are obtained all faces are represented as a linear combination of the eigen faces

$$\text{Any face} = \sum_{i=1}^k w_i E_i \quad (4.5)$$

where w_i is the weight of the i^{th} eigen face and E_i is the i^{th} eigen face. This k dimensional weight vector corresponds to the new representation of face in the PCA subspace. The weight vector is shown below.

$$W = [w_1 w_2 w_3 \dots w_k]^T \quad (4.6)$$

Since the Eigen vectors are Eigen faces, it will be efficient in representing face images. i.e. any face image can be defined as a Linear combination of Eigen faces. For non faces this is not the case as we are trying to represent a non face as a linear combination of faces. So the error will be high when we try to find a representation for non-faces as a linear combination of eigen faces. An example for eigen faces in spatial domain is given below.



FIGURE 4.1: First 10 eigen faces of the ORL Database of faces [33]

4.1 Stages in the development of the system

The important stages in the development of the system is given below

- Training Phase
 - Selection of training data
 - Representation in Curvelet Domain
 - Development of the training algorithm
 - Determination of parameters(threshold, eigen vectors)
- Testing Phase
 - Performance Analysis

4.1.1 Training Phase

During this phase the system is trained with the available training data based on the training algorithm. The training in the face detection system consists of face

images and non-face images. Face images can be obtained from any of the available online face databases. ORL[33] & Yale B[35] database is used in this work

4.1.1.1 Selection of training data

This is an important stage as the performance of the system depends on the training data. The different parameters which are considered for the selection of training data are given below.

TABLE 4.1: Parameters for selecting the training data

Parameter	Variations
Image Type	Color image, Grayscale image
Orientation of face	Front facing, profile rotated etc
Facial expression	Neutral expression, Varying expression
Background	Image with background, Cropped face only image

The selected parameters values are

- Grayscale image, Front facing, Neutral expression, Cropped face only region



FIGURE 4.2: Faces from Yale B extended database[35]

4.1.1.2 Representation of image in curvelet domain

The curvelet transform of the image is first calculated. The curvelet transform coefficients of one half is the complex conjugate of the other half. So any one of the half bands can be used for the image representation. The output of curvelet transform will contain coefficients corresponding to different scale and orientations. The coefficient columns at each scale and orientations are stacked one after the other to get a vector representation of the data in the curvelet domain.



FIGURE 4.3: Faces from ORL database of faces [33]

4.1.1.3 Training Algorithm

The basic flow chart of the system which is implemented is shown in figure 4.4

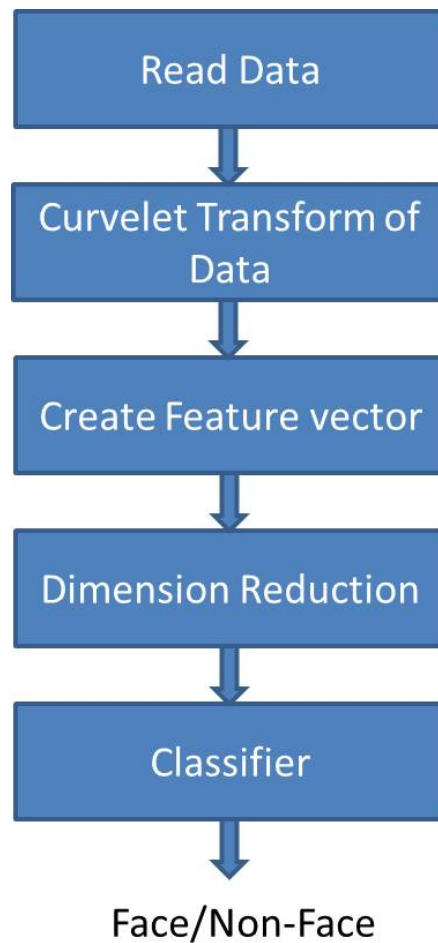


FIGURE 4.4: Flowchart

1. Read the training images and convert each image into column vectors in the curvelet domain. Form the matrix B(as given in equation 4.2) which consists of all these images.
2. Calculate the average face(Ψ) from the training data.

This average face is stored in the system.

3. Subtract the average face from all the training data and find the matrix A(equation 4.3).
4. Calculate the covariance matrix($C = AA^T$)
5. Calculate the Eigen values and Eigen vectors of the covariance matrix.

Direct calculation of eigen vectors is not possible in MATLAB as the dimension of the image vector is very large. So an indirect method is used to find the eigen values and eigen vectors. It is described as follows

Let A be the mean subtracted training data as described in equation 4.3.

$$A^T A u_i = \lambda_i u_i \quad (4.7)$$

$$(AA^T)(A u_i) = \lambda_i A u_i \quad (4.8)$$

$$A u_i = v_i \quad (4.9)$$

$$AA^T v_i = \lambda_i v_i \quad (4.10)$$

λ_i is the eigen value

u_i is the eigen vector of $A^T A$ and v_i is the eigen vector of AA^T

From the above equations it is clearly visible that the eigen vectors of AA^T is A times the eigen vectors of $A^T A$. The eigen vectors of AA^T is calculated from the eigen vectors of $A^T A$ which has got a dimension of L*L which is less than the dimension of covariance matrix which is MN*MN.

6. Order the eigen vectors in the decreasing order of the eigen values
7. Calculate the contribution of each eigen vectors

$$\text{Contribution of } i^{\text{th}} \text{ eigen vector} = \frac{\alpha_i}{\sum_{i=1}^k \alpha_i} \quad (4.11)$$

α_i is the eigen value of the i^{th} eigen vector.

8. Decide the total number of the eigen vectors to be kept to form the PCA sub space.

Sum of the contribution of all the eigen values will be one.

$$\sum_{i=1}^k \frac{\alpha_i}{\sum_{i=1}^k \alpha_i} = 1 \quad (4.12)$$

For eigen vectors with small eigen value the contribution will be very small. Consider the smallest number of eigen vectors corresponding to the largest eigen values which provides a high value for the sum of the contribution of eigen vectors. OR, alternately neglect the eigen vectors whose contribution is low compared to the contribution of the eigen vector with highest eigen value. These eigen vectors are normalized and is stored in the system.

9. Project the mean subtracted training images into the PCA subspace

Obtain the k-dimensional representation corresponding to each of these vectors. These vectors are stored in the system. This forms the lower dimensional representation of the original images. These feature vectors are stored in the system.

10. Calculation of threshold.

Find the average Euclidean distance between the k-dimensional representations of all the training faces to decide the threshold

4.1.1.4 Classification procedure

1. Given test image is converted into the vector format in curvelet domain.
2. Normalization.

Subtract the average face from the vector obtained in step 1.

3. Project the vector obtained in step2 to the PCA subspace and obtain the k-dimensional representation. Use the normalized eigen vectors stored in the system for projection.
4. Find the average distance value of this with the feature vectors stored in the system.

If distance is $<$ threshold then the input image is a face. Otherwise it is a non face.

4.1.2 Testing Phase

4.1.2.1 Confusion matrix & ROC

Confusion matrix is used as a performance evaluation measure in the classification problem. Face detection is a binary classification problem where the object is identified is either face or non face. The confusion matrix for face detection problem is given below. For computing the confusion matrix we need to provide the ground truth information to the system. A data set consisting of faces and non-faces are selected. The filename and the class to which each file belongs is made available to the system. The system will classify all the images in the data set as per the training algorithm and uses the ground truth information to compute the different entries in the confusion matrix. The ground truth information can be a text file which holds the name of the images and the class to which it belongs as a list. For example 0 can be used to indicate images that belong to class face and 1 can be used to indicate non faces.

The system was trained using 37 face images obtained from Yale B database. The performance analysis is done using the confusion matrix. The testing dataset had 37 faces(obtained from Yale B database) and 50 non faces(obtained after a random search on web). A typical value of confusion matrix obtained is given below

$$\text{Confusion Matrix} = \begin{pmatrix} 0.9615 & 0.0385 \\ 0.3488 & 0.6279 \end{pmatrix}$$

The variation of TPR and FPR as the threshold value is varied is given in figures 4.5 & 4.6 respectively.

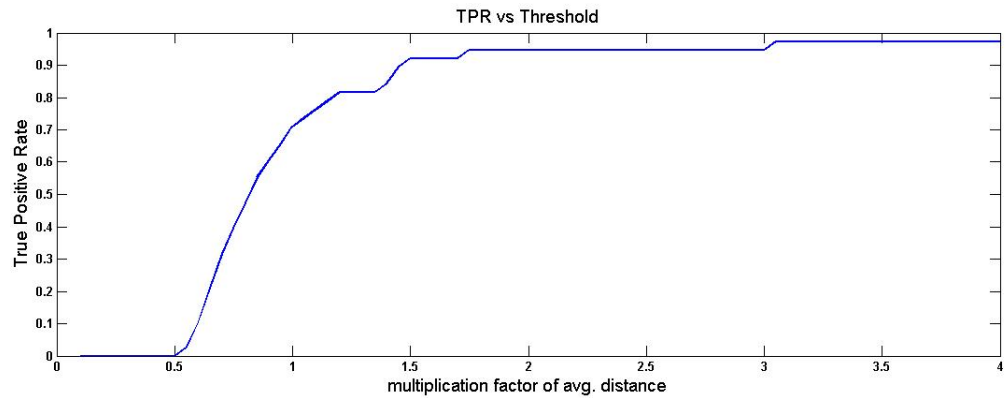


FIGURE 4.5: TPR vs Threshold

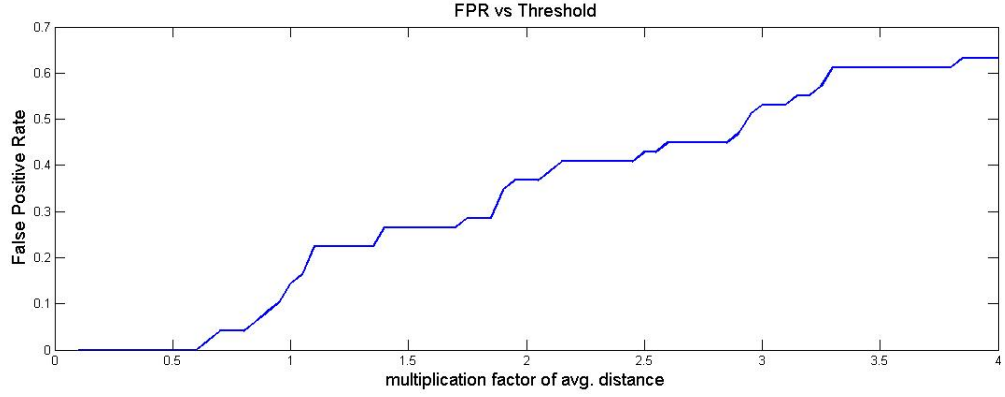


FIGURE 4.6: FPR vs Threshold

The receiver operating characteristic curve is shown in fig 4.7. The threshold distance is used is 1.6 times the average distance obtained during threshold calculation.

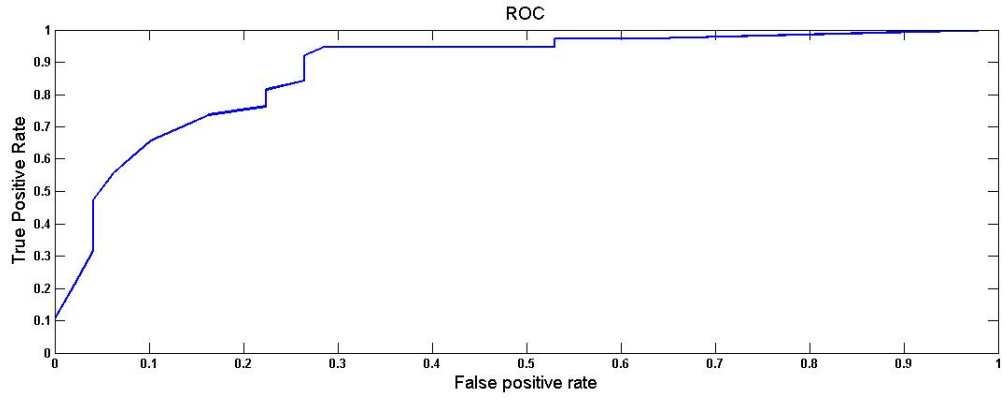


FIGURE 4.7: ROC

4.2 PCA directly on image intensity values

For comparing the performance of the curvelet domain based system, a system that works directly on the image intensity values is implemented. The performance of such a system is given below.

4.2.1 Performance

A typical value of the confusion matrix obtained is given below.

$$\text{Confusion Matrix} = \begin{pmatrix} 0.923 & 0.076 \\ 0.232 & 0.767 \end{pmatrix}$$

The values were obtained after training the system with 37 images from Yale B database. The performance was evaluated on a dataset which had 26 cropped face images obtained from ORL database and 43 non face images obtained from internet

4.2.2 Effect of threshold

If the threshold value is increased true positive rate increases, but at the same time false positive rate also increases. As the decision boundary is increased, more faces are detected and also many non-faces will fall inside the decision boundary. This increases both the TPR and FPR. So the threshold calculation should take this factor into consideration. The graph showing the effect of threshold on TPR and FPR is given below. The graph is obtained after training the system with 37 face images from Yale B database. The testing is done with a dataset of 37 images from Yale B database and 50 non faces obtained by random search from the internet. The threshold used here is a multiple of the average euclidean distance between the feature vectors of the 37 images obtained during the training.

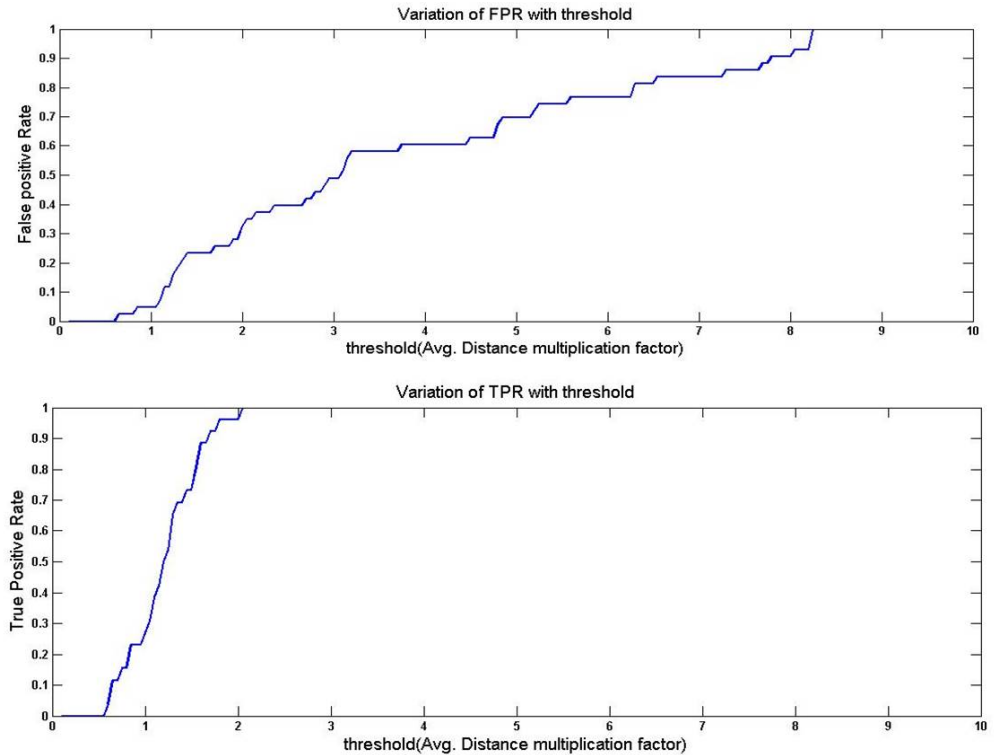


FIGURE 4.8: FPR vs threshold and TPR vs threshold

The receiver operating characteristic curve is shown in figure 4.9. The threshold distance used is 1.6 times the average distance obtained during threshold calculation.

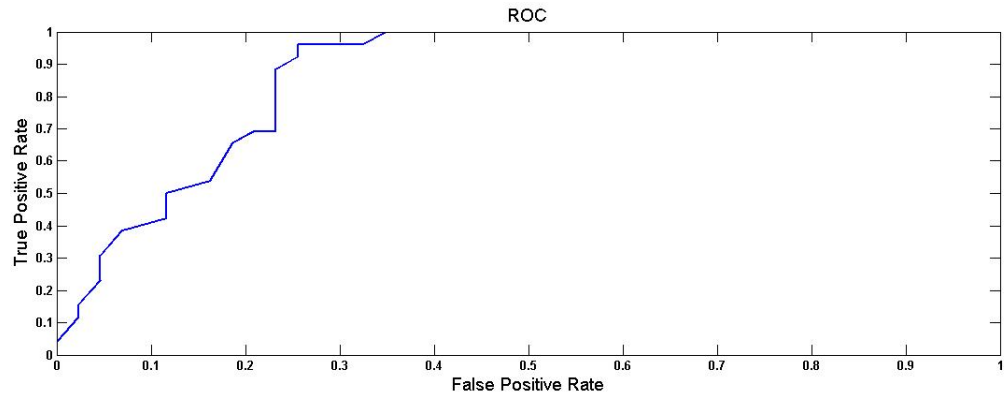


FIGURE 4.9: ROC

4.2.3 Effect of no. of eigen vectors

As the number of eigen vectors used to represent the PCA subspace is increased the accuracy of the system increases upto a point. After this increasing the number of eigen vectors doesn't increase the performance. When the number of eigen values are increased to the maximum it was found that the performance of the system went down. This is due to the fact that the eigen vectors with very low eigen value actually represents noise and not the data. So when this noise term is also considered the performance of the system decreases. The variation of the TPR with respect to the number of eigen values selected is given in figure 4.10.

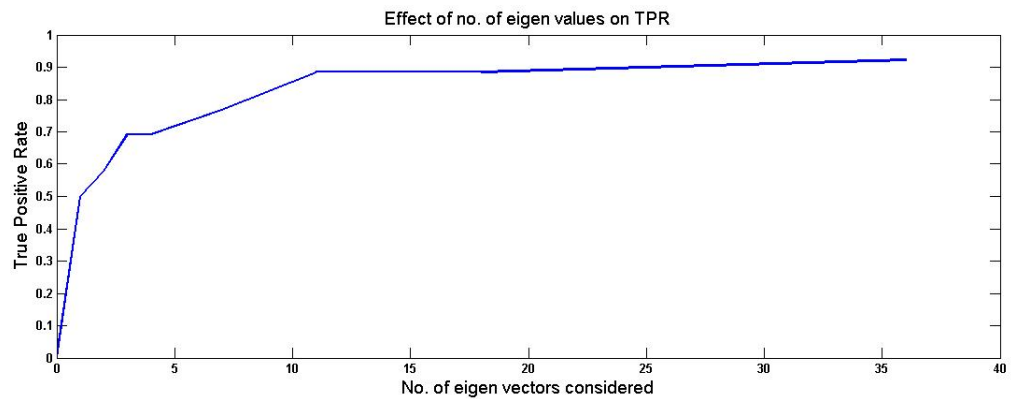


FIGURE 4.10: Variation of TPR with number of eigen vectors used

4.3 Observation

4.3.1 Eigen values of mean subtracted data

Let A be the mean subtracted data. Since it is a set of mean subtracted data its mean should be zero. So the sum of the column vectors in A should be result in zero vector.

$$a_1 + a_2 + a_3 + \cdots + a_N = 0 \quad (4.13)$$

$$a_1 = -(a_2 + a_3 + \cdots + a_N) \quad (4.14)$$

Above equation indicates that one of the column vector can be expressed as a linear combination of the other column vectors. This indicates that the columns are not linearly independent. If we calculate the rank of the matrix it will be $N-1$. i.e. one less than the no of column vectors. The covariance matrix is defined as $C=AA^T$. Since $\text{rank}(A)=N-1$ the rank of C will be also $N-1$. Since the rank of C is $N-1$ its determinant will be zero. A determinant can be calculated by taking product of the eigen values of the matrix. Since the determinant is zero one of the eigen values should be zero. This indicates that that corresponding eigen vector has no contribution in the data. This situation occurs when the number of training faces is less than the dimension of the data. i.e matrix A will have more number of columns compared to rows and all the columns are linearly independent.

4.4 Performance Comparison

TABLE 4.2: Performance Comparison

Training Dataset	Testing Dataset	AUC(Spatial Domain)	AUC(Curvelet Domain)
Yale(37 Faces)	ORL+Non-Faces (26 Faces + 43 Non-Faces)	0.8648	0.8215
Yale(37 Faces)	Yale+Non-Faces (37 Faces + 50 Non-Faces)	0.8939	0.8786
ORL(26 Faces)	ORL+Non-Faces (26 Faces+ 43 Non-Faces)	0.9534	0.9505
ORL(26 Faces)	Yale+Non-Faces (37 Faces + 50 Non-Faces)	0.7894	0.7801

The accuracy of the system which works in the curvelet domain is compared with the system which works directly on image intensity values. Area under the curve(AUC) of the receiver operating characteristic(ROC) curve is used as the comparison parameter. The AUC values indicate that the performance is almost equal to the method which works directly on the image intensity values.

4.5 Graphical User Interface

The MATLAB implementation of the PCA based method which works directly on the image intensity values is shown in figure 4.11. The MATLAB GUI implementation of

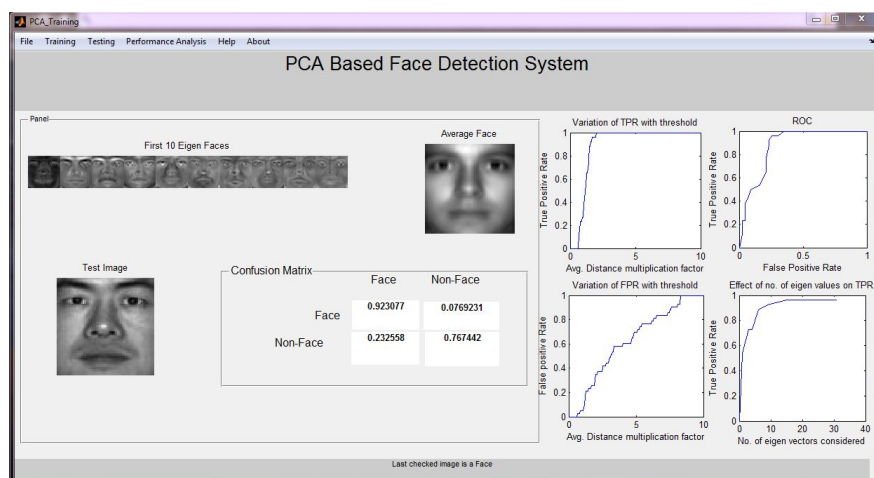


FIGURE 4.11: GUI of the system which works directly on intensity values

the system which works in the curvelet domain is shown in figure 4.12.

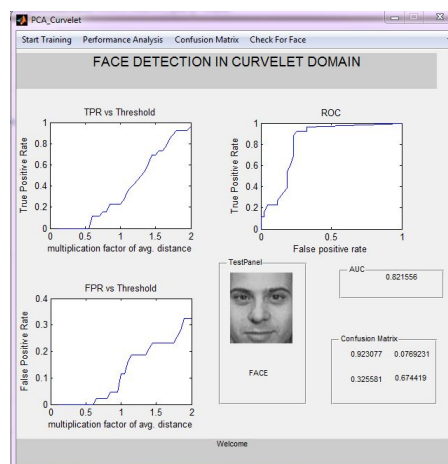


FIGURE 4.12: GUI of the system which works in curvelet domain

Chapter 5

Discussion and Future work

The existing face detection system was analyzed and a method for the development of a better system was proposed. The proposed method consists of curvelet transform for sparse representation of the data and a dimension reduction procedure using PCA. Curvelet transform was analyzed in detail in the work. The sparsity of curvelet transform was computed for different datasets. Another feature of the curvelet transform is the information content in the phase part of the transform. The information retained in the phase part of the transform is analyzed with the help of SSIM measure.

A representation of the features using curvelet transform was developed and analyzed. This representation had the curvelet coefficients at different scales and orientations. The performance was analyzed with the help of ROC and AUC. A comparison of the performance of the system using curvelet is done with a system which directly works on image intensities. AUC was used as the comparison parameter. The comparison revealed that the system based on the curvelet transform gave a similar performance in terms of accuracy with respect to the system which works on image intensities. This can be viewed from table 4.2. The reason behind the same level of accuracy can be understood only when an extensive analysis is done. This level of performance is not enough for practical applications. In order to have a better system a better representation of the features of the face using curvelet transform is required.

Appendix A

Background Mathematics

A.1 Covariance Matrix

Let X be a vector of dimension $n \times 1$.

$$X = [X_1 X_2 X_3 \dots X_n]^T \quad (\text{A.1})$$

The element at position i, j in the covariance matrix gives the covariance between the i^{th} and j^{th} elements in the vector X . The covariance matrix is defined as

$$\Sigma = \begin{pmatrix} E(X_1 - \mu_1)(X_1 - \mu_1) & E(X_1 - \mu_1)(X_2 - \mu_2) & \dots & E(X_1 - \mu_1)(X_n - \mu_n) \\ E(X_2 - \mu_2)(X_1 - \mu_1) & E(X_2 - \mu_2)(X_2 - \mu_2) & \dots & E(X_2 - \mu_2)(X_n - \mu_n) \\ \dots & \dots & \dots & \dots \\ E(X_n - \mu_n)(X_1 - \mu_1) & E(X_n - \mu_n)(X_2 - \mu_2) & \dots & E(X_n - \mu_n)(X_n - \mu_n) \end{pmatrix}$$

where $\mu_i = E(X_i)$ The diagonal elements is the variance of the i^{th} element in the vector X . When there is no redundancy in the data representation the covariance matrix will be diagonal.

A.2 Eigen vectors

Eigen vectors are those vectors when multiplied by a matrix retains its direction. Only the scaling of the vector will be changed. The scaling factor is called the eigen value.

The eigen vectors of a matrix A is defined by

$$Ax_i = \lambda_i x_i \tag{A.2}$$

where x_i is an eigen vector and λ_i is the corresponding eigen value. The equation shows that when an eigen vector is multiplied by the matrix A it is only scaled and it retains its direction. The eigen values are calculated using the equation

$$\det(A - \lambda I) = 0 \tag{A.3}$$

where I is the identity matrix.

Bibliography

- [1] Paul Viola and Michael Jones, “Rapid Object Detection using a Boosted Cascade of Simple Features,” in *CVPR Proc. IEEE Comput. Soc. Conf.*, 2001, pp. 511-518.
- [2] Yoav Freund and Robert E. Schapire, “A short Introduction to Boosting,” in *J. of Japanese Soc. for Artificial Intelligence*, 14(5):771-780, September, 1999
- [3] Ulrich Hoffmann, Jacek Naruniec, Ashkan Yazdani and Touradj Ebrahimi, “Face detection using discrete gabor jets and color information,” in *Int. Conf. on Signal Processing and Multimedia Applications*, 2008
- [4] Supratim Gupta, “Eye Image-Based Algorithms to Estimate Percentage Closure of Eye and Saccadic Ratio for Alertness Detection,” Ph.D. dissertation, Dept. Elec. Eng, IIT Kharagpur, WB, 2013.
- [5] Rafael C. Gonzalez and Richard E. Woods, *Digital Image Processing*, 2nd ed. Prentice Hall, 2009.
- [6] Steven W. Smith, *The Scientists and Engineer’s Guide to Digital Signal Processing*, 2nd ed. California Technical Publishing, 1999
- [7] Supratim Gupta, Anirban Dasgupta and Aurobinda Routray, “Analysis of Training Parameters for Classifiers Based on Haar-like Features to Detect Human Faces,” in *Int. Conf. on Image Information Processing*, 2011
- [8] Michael Elad, “Uniqueness and Uncertainty,” in *Sparse and Redundant Representations*, 1st ed. Springer, 2010.
- [9] Paul Viola and Michael Jones, “Fast multi-view face detection,” *Mitsubishi Electric Research Laboratories*, TR2003-96, Aug. 2003.
- [10] Yonina C. Eldar and Gitta Kutyniok, “Face Recognition by sparse representation” in *Compressed sensing*, 1st ed. Cambridge University press, 2012
- [11] J.L. Starceck, Fionn Murtagh and Jalal M. Fadili, “The Ridgelet and Curvelet Transform” in *Sparse Signal And Image Processing*, 1st ed. Cambridge University Press, 2010.

- [12] R. Lienhart, and J. Maydt, "An extended Set of Haar-like features for Rapid Object Detecton," *ICIP02*, pp.1:900-903, 2002
- [13] H. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. In *IEEE Patt.Anal. Mach. Intell.*, Vol. 20, pp. 22-38, 1998
- [14] H. Schneiderman and T. Kanade. A statistical method for 3D object detection applied to faces and cars. In *International Conference on Computer Vision*, 2000.
- [15] E. J. Candès and D. L. Donoho. New tight frames of curvelets and optimal representations of objects with piecewise- C^2 singularities. *Comm. on Pure and Appl. Math.* 57, 219–266, 2004.
- [16] E. Candes, L Demanet, D Donoho, L Ying "Fast discrete curvelet transforms" *SIAM: Multiscale Modeling & Simulation*, vol. 5, no. 3, pp. 861-899, 2006.
- [17] E. Candes and D. Donoho, "Curvelets - a surprisingly effective nonadaptive representation for objects with edges," in *Int. Conf. on Curves and Surface, Saint-Malo*, 1999.
- [18] Emmanuel Candes, David Donoho, Laurent Demanet, "Methods for performing fast discrete transforms of data," U.S. Patent 7 840 625 B2, Nov. 23, 2010.
- [19] Ma, Jianwei, and Gerlind Plonka, "A review of curvelets and recent applications," *IEEE Signal Processing Magazine*, vol.27, no.2, pp. 118-133, 2010.
- [20] I. Sumana, "Image retrieval using discrete curvelet transform," M.S.thesis, Monash University, Australia, 2008.
- [21] Sirovich, Lawrence, and Michael Kirby, "Low-dimensional procedure for the characterization of human faces", *J. of the Optical Society of America*, vol.4, no.3 ,pp.519-524, 1987.
- [22] Etemad, Kamran, and Rama Chellappa. "Discriminant analysis for recognition of human face images", *J. of the Optical Society of America*, vol 14, no.8, pp. 1724-1733, 1997.
- [23] I. Daubechies, *Ten lectures on wavelets*, 6th ed. Philadelphia, SIAM, 1992.
- [24] Kanade, Takeo, "Picture processing system by computer complex and recognition of human faces", 1974.
- [25] Shadi AlZubi, Naveed Islam, and Maysam Abbod, "Multiresolution Analysis Using Wavelet, Ridgelet and Curvelet Transforms for Medical Image Segmentation", *Journal of Biomedical Imaging*, 2011.

- [26] D. Lee Fugal, *Conceptual Wavelets in Digital Signal Processing*, 1st ed. San Diego, Space & Signals Technical Publishing, 2009.
- [27] Brian Eriksson, (2008). The Very Fast Curvelet Transform, [Online] Available : http://homepages.cae.wisc.edu/~ece734/project/s06/eriksson_rpt.pdf
- [28] A. Averbuch, R.R. Coifman, D.L. Donoho, M. Elad and M. Israeli, "Fast and accurate polar Fourier transform," *Applied and Computational Harmonic Analysis*, vol. 21, no. 2, pp. 145-167, 2006.
- [29] Kutyniok, Gitta, Morteza Shahram, and David L. Donoho, "Development of a digital shearlet transform based on pseudo-polar FFT," in *SPIE Optical Engineering + Applications*, 2009.
- [30] Laurent Demanet, Curvelet Transform, [Online] Available: www.curvelet.org
- [31] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience*, vol. 3, no. 1, 1991
- [32] M. Turk and A. Pentland, "Face Recognition using Eigenfaces," *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 1991, pp. 588-591.
- [33] AT&T Laboratories Cambridge, ORL Database of Faces. [Online] Available : <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>
- [34] Databases for Face Detection and Pose Estimation, [Online] Available : http://robotics.csie.ncku.edu.tw/Databases/FaceDetect_PoseEstimate.htm
- [35] K.C. Lee and J. Ho and D. Kriegman, "Acquiring Linear Subspaces for Face Recognition under Variable Lighting", *IEEE Transactions Pattern Analysis and Machine Intelligence*, vol. 27, pp. 1-15, 2005